

Practical Software and Systems Measurement User's Group - 2012



***Software Maintenance Life
Cycle Cost Estimation Model***

***Workshop Materials and Discussion
Topics***

***Jack McGarry - U.S. Army ARDEC
Dr. Robert Charette - ITABHI Corp.***

1 August 2012

Workshop Objectives

- ***Present an integrated software maintenance life cycle estimation model that links software maintenance resource requirements to planned program sustainment profiles***
- ***Examine the viability of the underlying model concepts and constructs***
- ***Gather inputs focused on improving the design and implementation of the model***

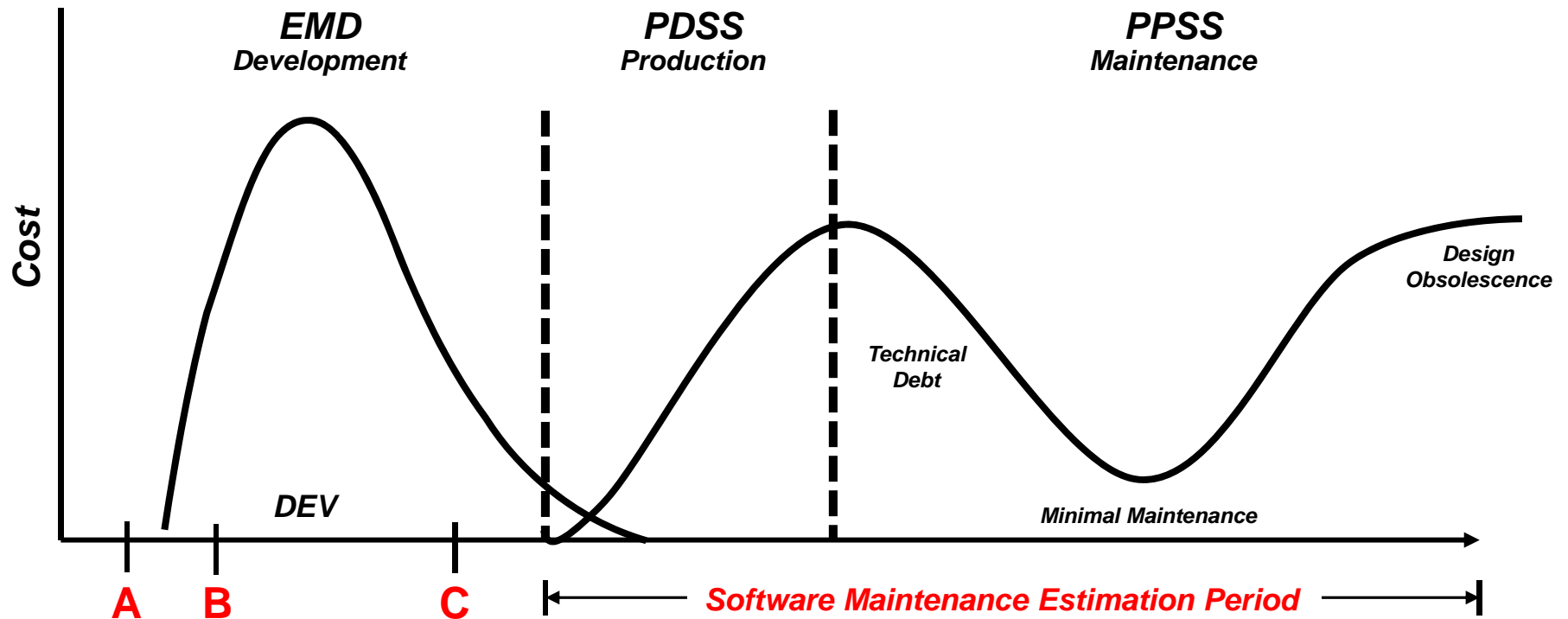
Participants

- *Brad Clark*
- *Denton Tarbet*
- *Cari Pullen*
- *Chris Miller*
- *Bob Epps*
- *Paul McMahon*
- *Marc Gutleber*
- *Kim Elliott*
- *Rich Mabe*
- *Larry Osiecki*
- *Don Beckett*
- *Bob Ferguson*
- *Corinne Wallshein*
- *Joe Dean*
- *Jeramia Poland*
- *Bob McCann*
- *Thomas Harless*
- *Rich Schubert*
- *Arlene Minkiewicz*

Workshop Agenda

- ***Review the overall estimation model requirements, architecture, and key concepts***
- ***Examine the underlying model constructs***
 - ***Underlying quantitative and context data***
 - ***Software maintenance products and activities***
 - ***Cost estimating relationships - cost drivers***
 - ***Risk and uncertainty application constructs***
 - ***Integrated cost allocation model***
- ***Summarize inputs and recommendations***

Notional Software Maintenance Life-Cycle Cost Model



Outcomes

- ***Program context fundamentally impacts the baseline cost estimation process***
 - ***Data structures and data availability***
 - ***Domain estimation relationships - norms***
 - ***Resource allocation - application structures***
- ***The software maintenance WBS needs to be adaptable to each program - and still provide a common task-activity baseline***
- ***Primary focus remains the release profile***
- ***Need to refine the design and mapping of the prospective CERs***

Problem Statement

Current software maintenance estimation methodologies do not support objective and defensible resource projections and allocations in the emerging Army systems sustainment environment

Practical Software and Systems Measurement User's Group - 2012



***Software Maintenance Life
Cycle Cost Estimation Model***

***Workshop Materials and Discussion
Topics***

***Jack McGarry - U.S. Army ARDEC
Dr. Robert Charette - ITABHI Corp.***

1 August 2012

Workshop Objectives

- ***Present an integrated software maintenance life cycle estimation model that links software maintenance resource requirements to planned program sustainment profiles***
- ***Examine the viability of the underlying model concepts and constructs***
- ***Gather inputs focused on improving the design and implementation of the model***

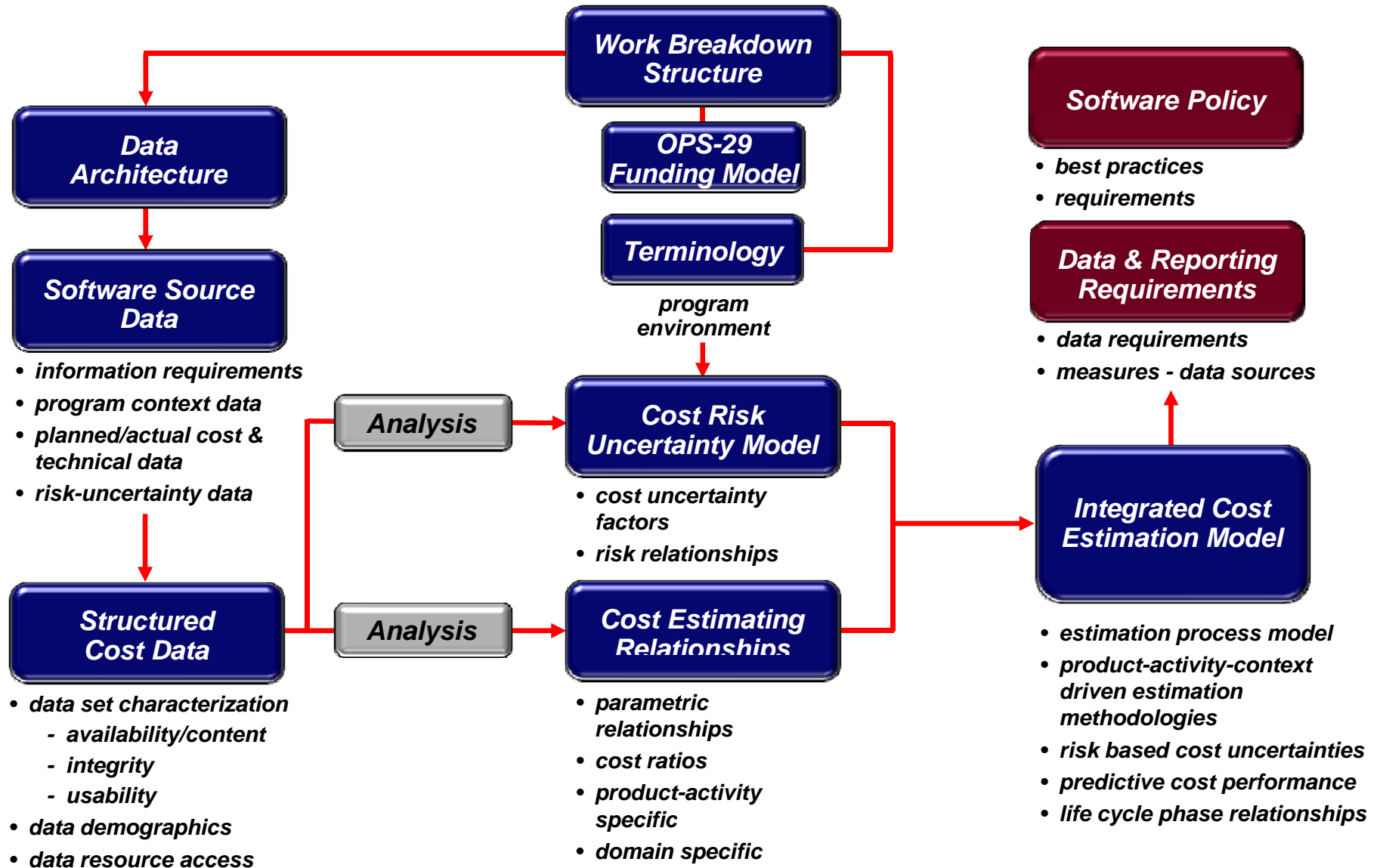
Workshop Agenda

- ***Review the overall estimation model requirements, architecture, and key concepts***
- ***Examine the underlying model constructs***
 - ***Underlying quantitative and context data***
 - ***Software maintenance products and activities***
 - ***Cost estimating relationships - cost drivers***
 - ***Risk and uncertainty application constructs***
 - ***Integrated cost allocation model***
- ***Summarize inputs and recommendations***

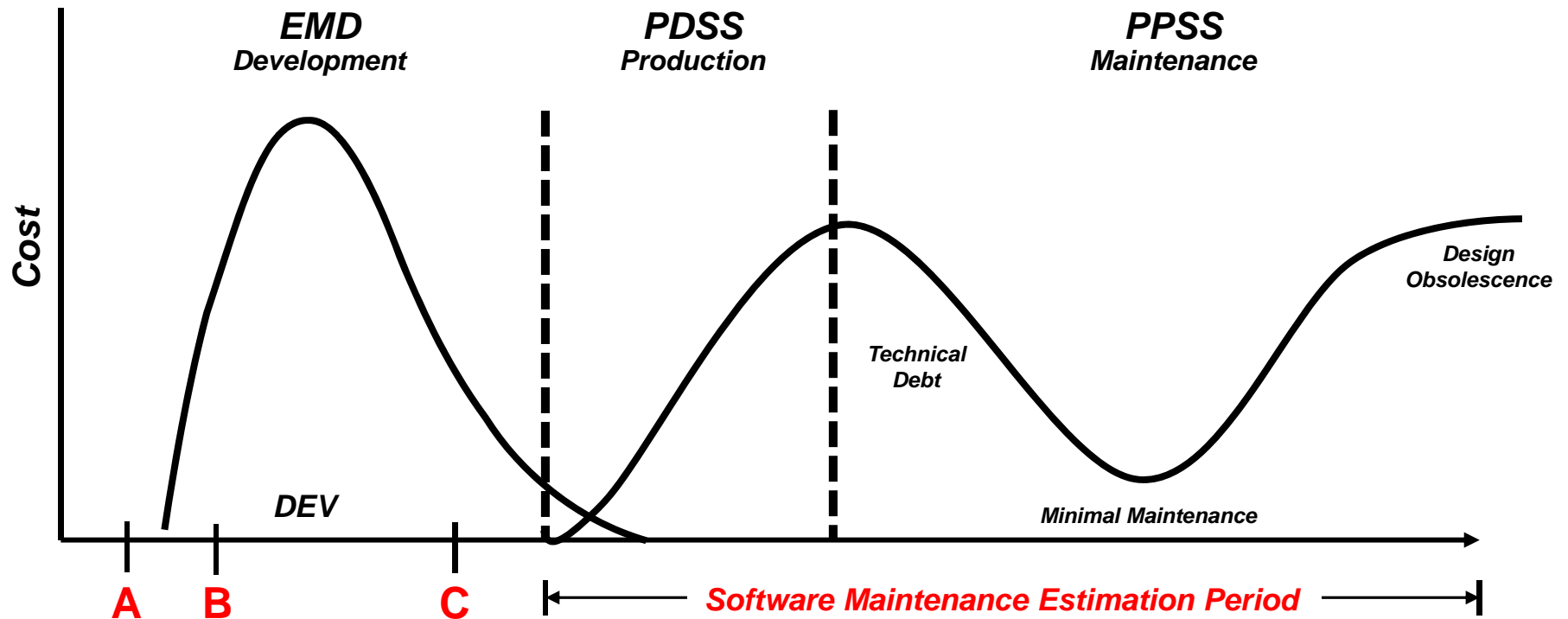
Estimation Methodology Requirements

- ***Integrate software maintenance product and activity costs over the life cycle of a program***
- ***Align the cost profile to the specific program life cycle software maintenance improvement and release strategy***
- ***Encompasses both PDSS and PPSS efforts***
- ***Ability to estimate at Milestones A, B, and C***
- ***Integrates component cost estimates from multiple CER based product and activity estimates (WBS defined)***
- ***Directly related to development technical parameters and development performance***
- ***Can incorporate cost risk uncertainty at component and higher levels***
- ***Practical - defensible - data derived - adaptable***

Practical Software and Systems Measurement



Notional Software Maintenance Life-Cycle Cost Model



Current Software Maintenance Estimation Methods

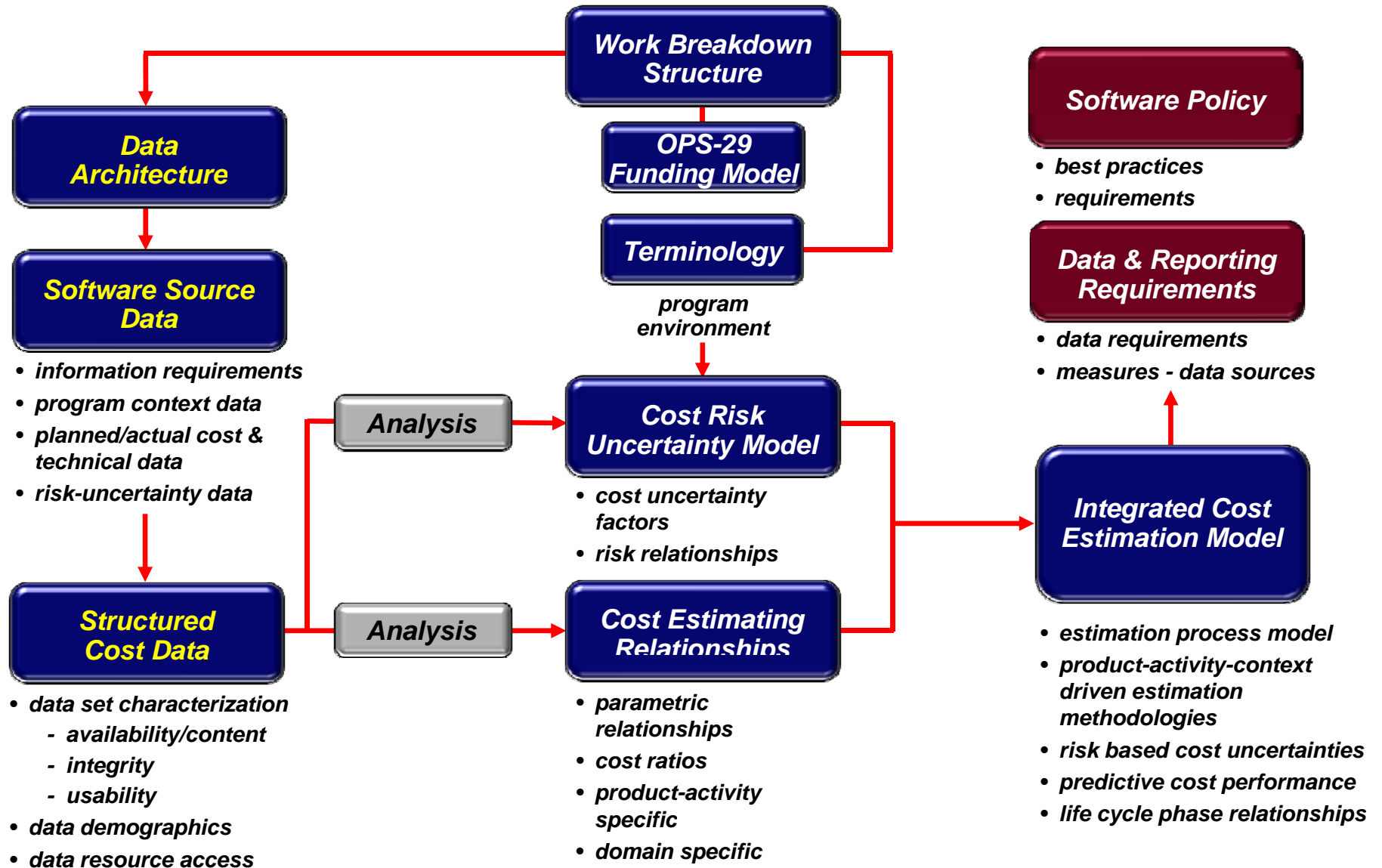
- ***Parametric models***
 - ***Product focused - only include corrective, perfective, adaptive changes & enhancements***
 - ***Invalid assumptions about sustaining engineering tasks***
 - ***Historical data not visible in all models***
 - ***Not calibrated - validated***
- ***Historical software maintenance funding - effort***
 - ***Limited access to historical FTE - Budget data-***
 - ***Represent “rolled-up” costs***
 - ***Requirements or LOE funding?***
 - ***Availability of correlated program development - sustainment cost data***

Current Estimation Methods (continued)

- ***Number of lines of code per software engineer***
 - *Each engineer can maintain 20K-25K LOC/ESLOC*
 - *Does not reflect the impact of software reuse or COTS*
- ***Software maintenance estimated as a percentage of development costs***
 - ***Rule(s) of thumb - development based:***
 - *S/W maintenance costs - 2/3 of total S/W life cycle costs*
 - *S/W maintenance costs - 60% to 75% of total S/W life cycle costs*
 - *Annual S/W maintenance costs - 5% to 10% of total S/W life cycle costs*
 - *Ignores total system life cycle software growth and maintenance requirements/strategy/tasks*

***All of these methods have significant limitations
in the current Army environment***

Practical Software and Systems Measurement



Categories of Data

- **Quantitative performance data - effort, product measures, activity measures, etc.**
- **Funding data - budgets, expenditures, dollars and resource allocations, accounting structures, etc.**
- **Technical data - domain, mission, architecture, interfaces, etc.**
- **Program profile data - maintenance/release strategy, schedules, no. of deployed systems, etc.**
- **Program context data - technical and operational risks, environmental risks, mandates, policy, etc.**

Data is Time Dependent

Operating Environments - Domains

Count of Operating Environment					
	Helicopter	UAV	Ground	Missile	Grand Total
Mission Processing			5		5
Real Time Embedded - Other	3		5		8
System			4		4
Training			6		6
Vehicle Control		2		2	4
Vehicle Payload			2		2
Grand Total	3	2	22	2	29

Count of Super-Domain					
	Helicopter	UAV	Ground	Missile	Grand Total
Mission Critical - Embedded	3	2	7	2	14
Mission Critical - Non-Embedded			9		9
Mission Support			6		6
Grand Total	3	2	22	2	29

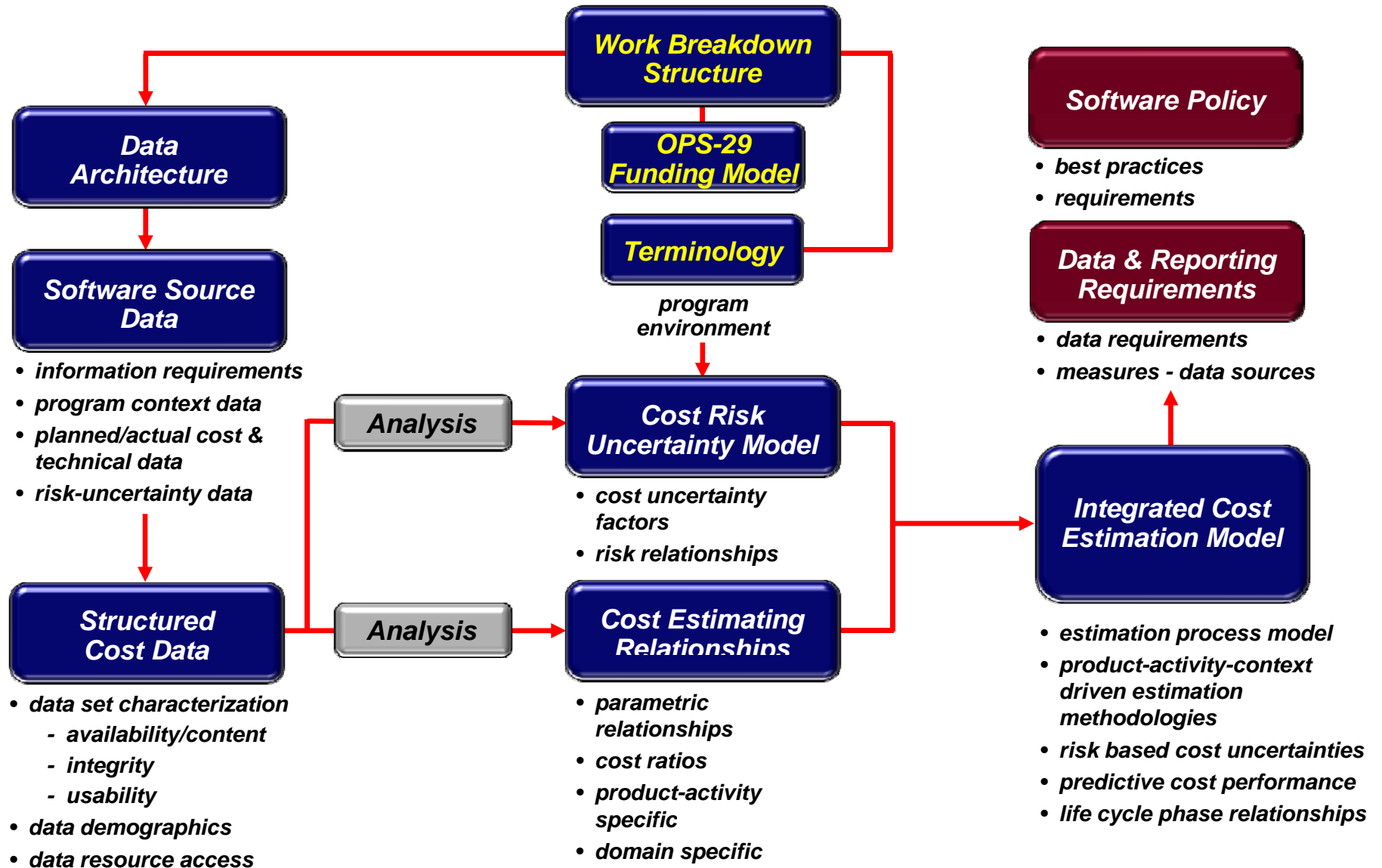
Army Software Maintenance Data

		Number of Releases	WBS Task Performed	Interfaces	Effort	Cost (\$)	Schedule	Software Change Requests (SCRs)	Requirements	Defects	Backlog	Source Lines of Code	OPS-29 Data
Org 1	Program 1	3											
	Program 2	1											
	Program 3	1											
	Program 4	4											
	Program 5	1											
	Program 6	3											
	Program 7	1											
	Program 8	1											
	Program 9	1											
	Program 10	4											
	Program 11	4											
	Program 12	1											
	Program 13	1											
Org 2	Program 1	2											
	Program 2	2											
	Program 3	5											
	Program 4	3											
	Program 5	1											
	Program 6	4											
	Program 7	2											
	Program 8	2											
	Program 9	1											
Org 3	Program 1	1											
Org 4	Program 1	6											
	Program 2	2											
	Program 3	3											
	Program 4	7											
Org 5	Program 1	Multiple											
	Program 2	Multiple											
	Program 3	Multiple											
Org 6	Program 1	3											
	Program 2	6											
	Program 3	3											
	Program 4	8											
	Program 5	1											
		88											

Data Analysis Summary

- ***Availability of data varies significantly by organization and program***
- ***Execution data (actuals) is very limited - most data was planning data aligned with the OPS-29 funding model***
- ***Multiple funding streams exist for similar work***
- ***Data is cost based - not product/activity based***
- ***Software engineering data derived from different source artifacts for a given release are sometimes inconsistent***
- ***The aggregation levels of provided data are inconsistent (details vs. summary)***
- ***The data that was available provided significant insight into potential cost estimating relationships and maintenance products and activity allocations***

Practical Software and Systems Measurement



Work Breakdown Structure

**Software Maintenance, Sustaining Engineering,
and Operational Support**



Version 2.82

Work Breakdown Structure

- ***Common structure that includes all potential products and activities - “what’s in - what’s out”***
- ***Common definitions - terminology***
- ***Emphasis on DOD weapons systems***
- ***Basis for identifying specific cost elements attributable to a given program or system maintenance/sustaining engineering effort***
- ***Product based - driven by changes to the software baseline(s)***
- ***Release focused - primary software maintenance product***
- ***Intended to be tailored and adapted for each program or organization:***
 - ***addition/deletion of lower level cost elements***
 - ***re-binning of lower level cost elements***
- ***Foundation for cost estimation process/models***
- ***Basis for defining cost estimating relationships***

Work Breakdown Structure

- 1.0 Software Maintenance - products and activities associated with modifying an operational software product or system**
- 2.0 Software Licenses - products and activities associated with the procurement and renewal of software licenses for operational software**
- 3.0 Information Assurance - products and activities associated with ensuring that the software is compliant with externally defined information assurance requirements**
- 4.0 Certifications and Accreditations - products and activities associated with verifying a software system against externally defined domain performance criteria**
- 5.0 Sustaining Engineering - products and activities associated with supporting a deployed software product or system in its operational environment**
- 6.0 Facilities & Infrastructure - products and activities associated with establishing and operating the facilities and processes required to modify, integrate, and test operational software products or systems**
- 7.0 Management - products and activities associated with planning, organizing, funding, and controlling the resources required to support operational software products or systems**

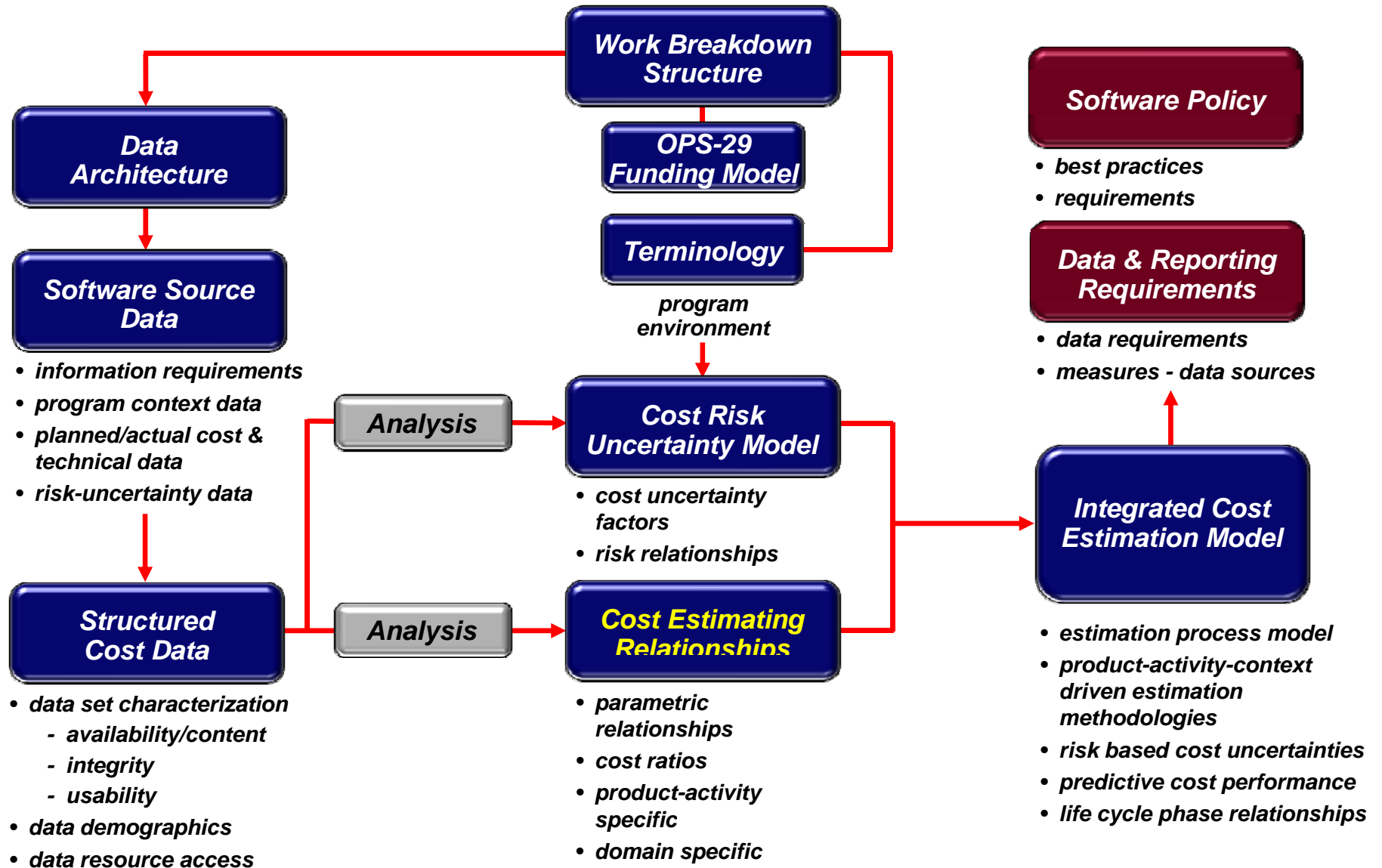
Program WBS Element Coverage

WBS	Title	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1.0	Software Maintenance	17	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█												
1.1	Software Release (N)	7	█	█	█	█	█	█	█																							
1.1.1	Planning and Management	19	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
1.1.2	Software Requirements	24	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
1.1.3	Architecture and Design	24	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
1.1.4	Change Implementation	23	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
1.1.4.1	Change Development	7	█	█	█	█	█	█	█																							
1.1.4.2	Unit Testing	7	█	█	█	█	█	█	█																							
1.1.5	Integration and Test	24	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
1.1.5.1	Test Planning	8	█	█	█	█	█	█	█																							
1.1.5.2	Test Scenario and Test Case Development	8	█	█	█	█	█	█	█																							
1.1.5.3	Test Tool Development	8	█	█	█	█	█	█	█																							
1.1.5.4	Test Conduct	8	█	█	█	█	█	█	█																							
1.1.6	Acceptance Testing	29	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
1.1.6.1	Test Planning	7	█	█	█	█	█	█	█																							
1.1.6.2	Test Scenario and Test Case Development	7	█	█	█	█	█	█	█																							
1.1.6.3	Test Tool Development	7	█	█	█	█	█	█	█																							
1.1.6.4	Test Conduct	7	█	█	█	█	█	█	█																							
1.1.7	Rework	7	█	█	█	█	█	█	█																							
1.1.8	Emergency Repairs	25	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
1.1.9	Hardware Updates	15	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
1.2	Software Release (N+1)	0																														
1.3	Software Release (N+2)	0																														
1.4	Software Release (N+3)	0																														
2.0	Licenses	11	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
3.0	Information Assurance	27	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
3.1	IAVA	12	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
3.2	DIACAP	11	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
4.0	Certifications & Accreditations	12	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	

Program WBS Element Coverage (cont.)

5.0	Software Sustaining Engineering	0																				
5.1	Analysis and Studies	18																				
5.2	Test Support	16																				
5.3	Delivery	15																				
5.4	User Training	15																				
5.5	User Support	16																				
5.6	Field Support	26																				
6.0	Software Facilities and Infrastructure	16																				
6.1	Development Facilities	11																				
6.1.1	Equipment	7																				
6.1.2	Licenses	11																				
6.1.3	Operations	7																				
6.2	Integration and Test Facilities	27																				
6.2.1	Equipment	7																				
6.2.2	Licenses	14																				
6.2.3	Operations	11																				
6.4	Tactical Equipment	11																				
6.5	Test Equipment and Tools	27																				
7.0	Management	11																				
7.1	Program Management	29																				
7.1.1	Project Release Management	12																				
7.1.2	Risk Management	11																				
7.1.3	Measurement/ Analysis	26																				
7.2	Contract Management	17																				
7.3	Change Management	27																				
7.4	Data Management	24																				
7.5	Quality Assurance	29																				
7.5.1	IV&V	26																				
7.6	Process Management	12																				
7.7	Personnel Management	0																				

Practical Software and Systems Measurement



Cost Estimating Relationships

- ***Types of CERs***
 - ***product***
 - ***activity***
- ***CER constructs***
 - ***parametric***
 - ***cost ratios***
 - ***simple relationships***
 - ***historical norms***
- ***Types of costs***
 - ***variable***
 - ***semi-variable***
 - ***fixed***
- ***Key considerations***
 - ***domain***
 - ***life cycle phase***
 - ***change drivers***

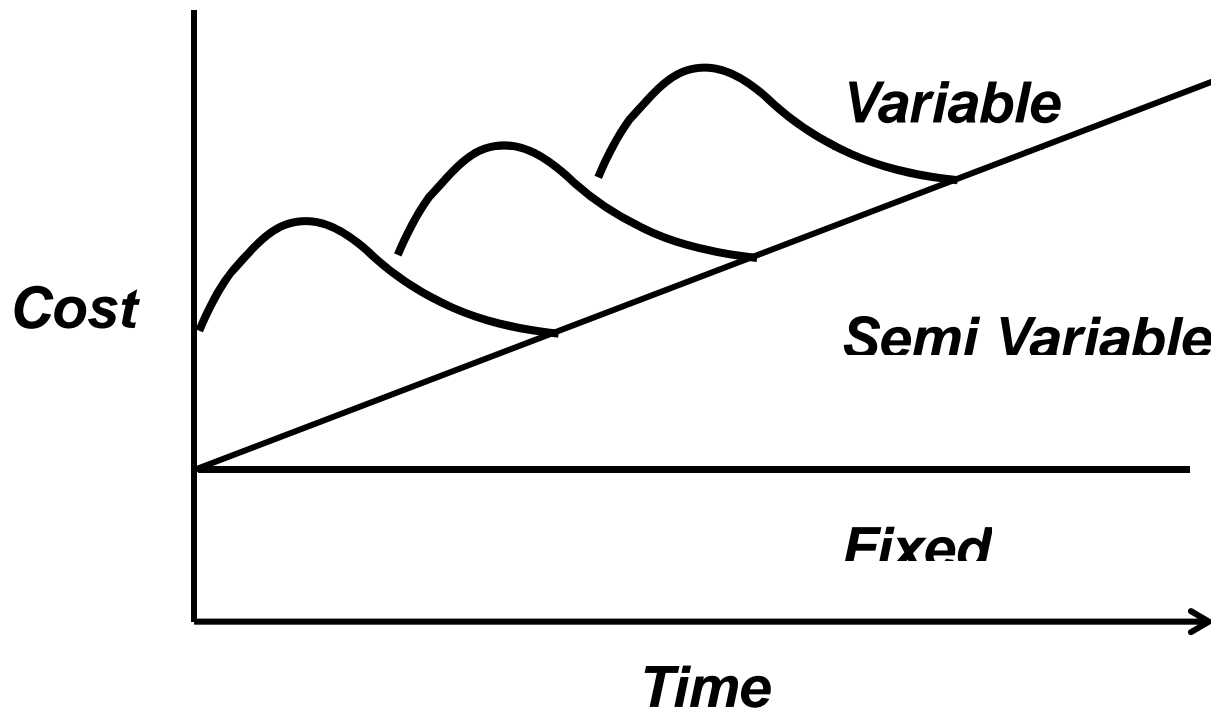
Cost Estimating Relationships

- ***Ratios used to identify performance relationships***
- ***Product costs structured into software releases - estimated parametrically based on functional change content***
- ***Activity costs estimated using simple CER relationships - unique variables and drivers***
- ***Fixed costs based on historical resource expenditures***
- ***Related factors influence and/or modify the outcomes in all cases***

WBS Based Cost Estimating Relationships

WBS	Title	Army OPS-29 Mapping	Variability	Cost Drivers Examples
1.1	Software Release			Lower level cost roll-up
1.1.1 to 1.1.5	Planning, Req'ts, Design, Implementation, Development, Unit Testing, Integration & Test (typical SW Development)	Capability Sets FY(XX/XX) System Mission Capability	Variable Cost	Parametric modeling - Number of Requirements (Enhancements), Defects, Test Cases - added, reused, modified; App. Domain, Complexity, Reliability, Personnel Factors
1.1.6	Acceptance Test	Capability Sets FY(XX/XX) System Mission Capability	Variable Cost	Number of requirements / capabilities / ... / test cases / etc.)
2.0	Licenses	Licenses	Semi-Var	Type and number of COTS products.
3.1	IAVA	IAVA	Semi-Var Cost	Parametric model (percentage distribution); Application domain, size, effort staffing, duration, productivity
4.0	Certifications & Accreditations	C&A	Fixed	Cost per C&A
5.0	SW Sustaining Engineering		Roll-up	Lower level cost roll-up
5.1 to 5.5	Analysis, Test Support, Delivery, Training	Organic Labor	Semi-Var Cost	LOE
5.6	Field Support	FSEs	Semi-Var	LOE, No. of Field Sites
6.0	Software Facilities & Infrastructure	System Infrastructure or System Open Door	Fixed	# People at facility, Simulation / Test Equip Maintenance
7.0	Management	Organic	Roll-up	Lower level cost roll-up
7.1 to 7.7	Program / Release / Risk / Contract Management		Fixed Cost	LOE

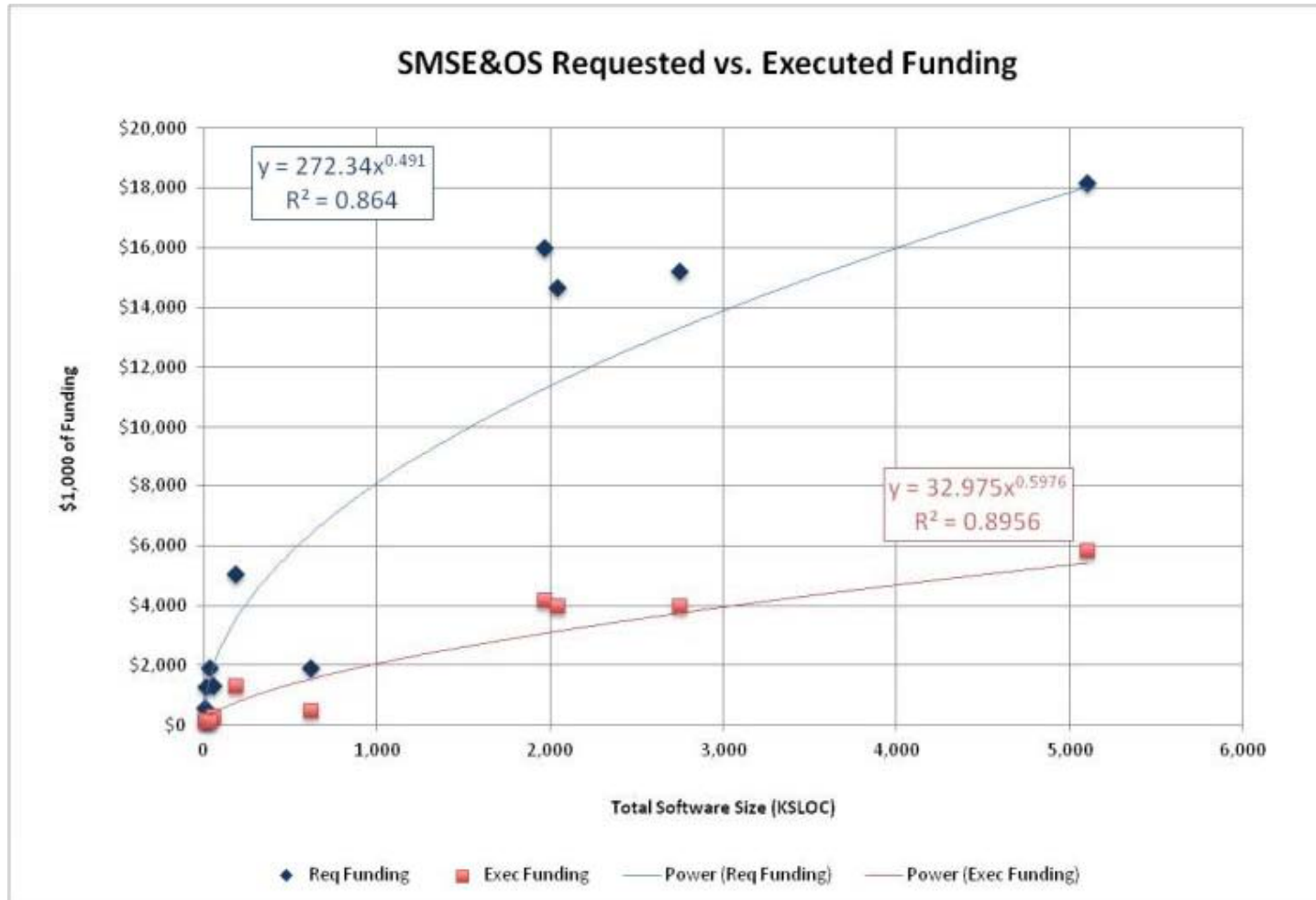
Types of Software Maintenance Costs



Correlation Coefficients Matrix

Correlation Coefficients Matrix						
Sample size		8	Critical value (10%)	1.94		
		Interfaces	Age	Est_Total_KSLOC	Req_Fund_FY11	Exec_Fund_FY11
Interfaces	Pearson Correlation Coefficient	1.00				
	<i>R Standard Error</i>					
	<i>t</i>					
	<i>p-value</i>					
	<i>H0 (10%)</i>					
Age	Pearson Correlation Coefficient	0.47	1.00			
	<i>R Standard Error</i>	0.13				
	<i>t</i>	1.31				
	<i>p-value</i>	0.24				
	<i>H0 (10%)</i>	accepted				
Est_Total_KSLOC	Pearson Correlation Coefficient	0.70	0.23	1.00		
	<i>R Standard Error</i>	0.08	0.16			
	<i>t</i>	2.43	0.58			
	<i>p-value</i>	0.05	0.58			
	<i>H0 (10%)</i>	rejected	accepted			
Req_Fund_FY11	Pearson Correlation Coefficient	0.39	0.30	0.87	1.00	
	<i>R Standard Error</i>	0.14	0.15	0.04		
	<i>t</i>	1.04	0.77	4.41		
	<i>p-value</i>	0.34	0.47	0.00		
	<i>H0 (10%)</i>	accepted	accepted	rejected		
Exec_Fund_FY11	Pearson Correlation Coefficient	0.50	0.59	0.81	0.94	1.00
	<i>R Standard Error</i>	0.13	0.11	0.06	0.02	
	<i>t</i>	1.41	1.78	3.40	6.87	
	<i>p-value</i>	0.21	0.13	0.01	0.00	
	<i>H0 (10%)</i>	accepted	accepted	rejected	rejected	

Funding vs. Software Size



Preliminary Cost Relationships

- ***Results***
 - ***Estimated total system KSLOC and number of interfaces had the highest correlation to both funded and executed funding***
 - ***Requested and executed funding varied by a factor of 3***
 - ***There is an implied prioritization of applied funding by subsystem (domain - weapons system vs. training system)***
 - ***System age had no appreciable cost impact***
 - ***There is a non-linear relationship between software system size and cost for software maintenance (resource constraints)***
 - ***Measurable software development characteristics available early in the life cycle may be early predictors of software maintenance life cycle cost***

Software Change Drivers

(Product CER Calibration Factors)

External Drivers

- **Operational User**
 - *Functional modifications*
 - *Functional additions*
 - *Functional deletions*
- **Stakeholder**
 - *Threat*
 - *Mission doctrine*
 - *System interoperability*
 - *External testing/IV&V*
 - *External audits*
- **Mandate**
 - *Legal/Regulatory/Policy*
- **Technology**
 - *Technology obsolescence*
 - *Infrastructure changes*

Internal Drivers

- **Legacy**
 - *Technical debt*
 - *Deferred functionality*
- **Maintenance Operations**
 - *Maintainer skill set*

Types of Changes

- **Corrective (identified defects)**
- **Preventive (latent faults)**
- **Perfective (functional enhancements)**
- **Adaptive (new HW/SW environments)**

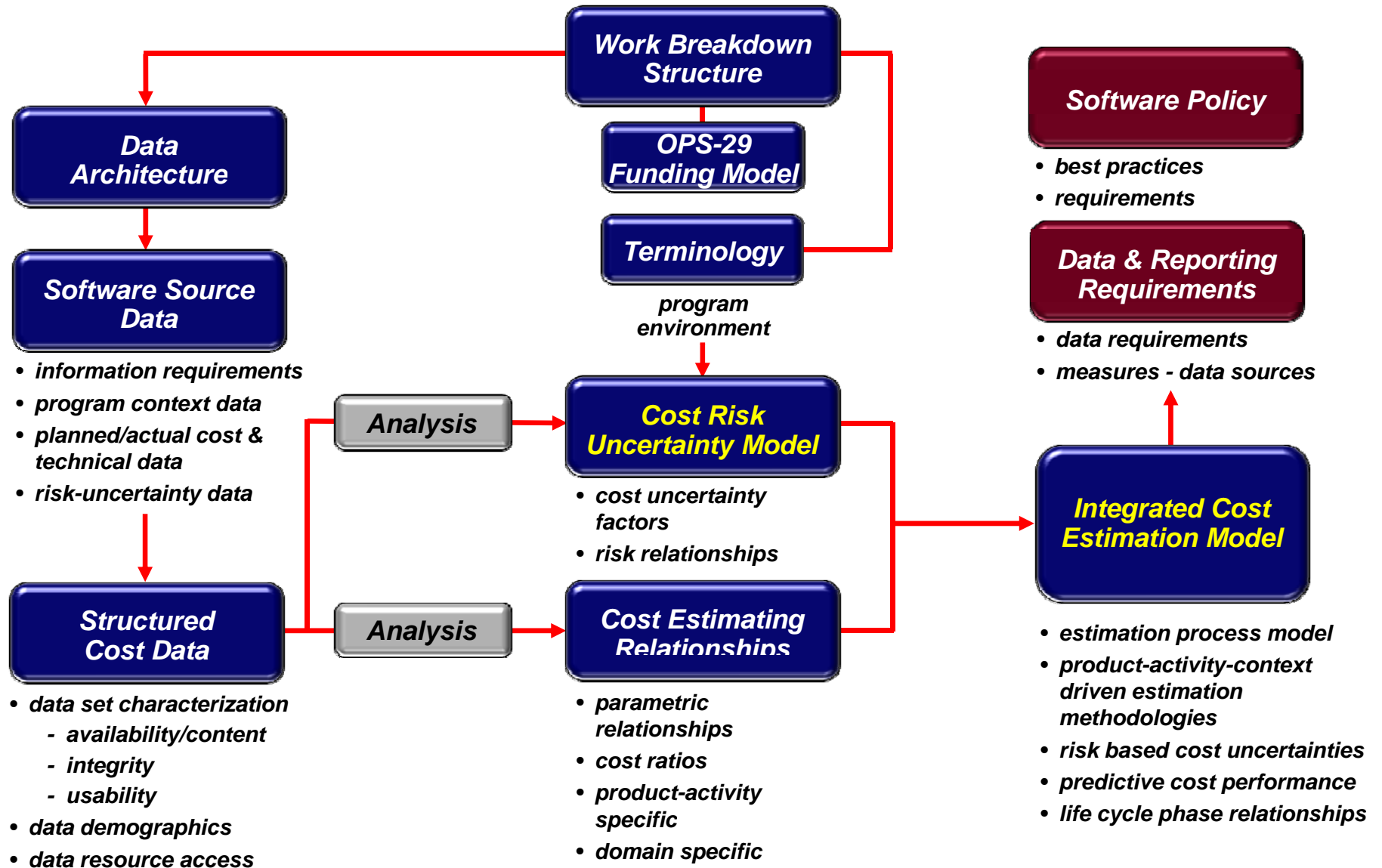
Software Maintenance Change Drivers

- ***The factors that cause the changes that are made to an operational software baseline***
- ***Software changes characterized by:***
 - ***source of change***
 - ***type of change***
 - ***impact of change (scope, complexity, etc.)***
 - ***priority of implementation***
- ***Change drivers can impact the operational software configuration, associated sustaining engineering activities, and the implemented maintenance infrastructure***
- ***Most change requirements are allocated to planned releases per time period***
- ***Deferred change requirements are defined as “backlog”***
- ***Drivers with different characteristics are costed differently***

Software Maintenance Product Sizing

- ***Overall concept of “E size” for maintenance products***
- ***E Size for types of changes (different drivers)***
- ***Composite E Size for different types of releases***
- ***Concept - profile the program maintenance strategy based on numbers and types of releases (variable costs) - add semi-fixed costs related to supporting the fielded software (costs not based on amount of work to construct a release)***
- ***Variable costs driven by type and number of product changes***
- ***Semi-fixed costs driven by number of deployed platforms, domain, operating tempo, etc.***
- ***Similar to software development sizing “E Sloc” but focused on change drivers/release content and support scope***

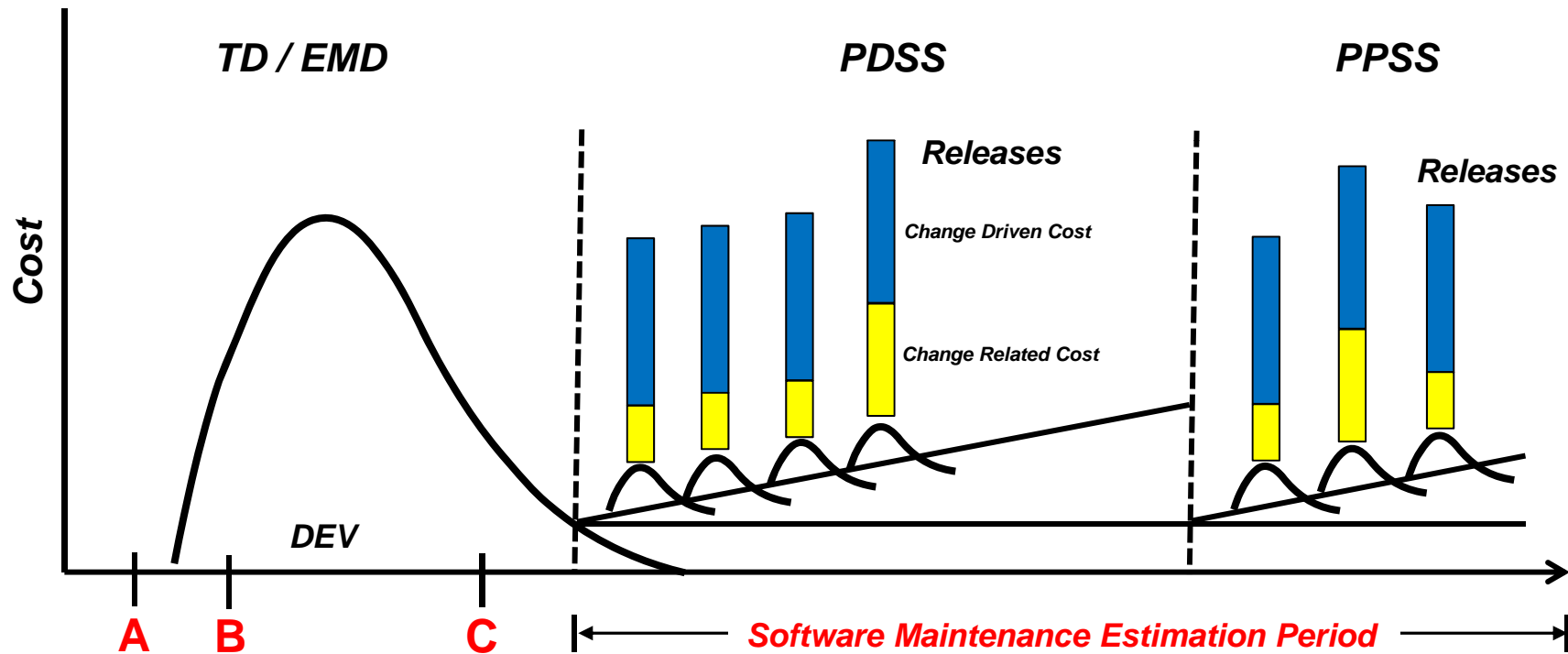
Practical Software and Systems Measurement



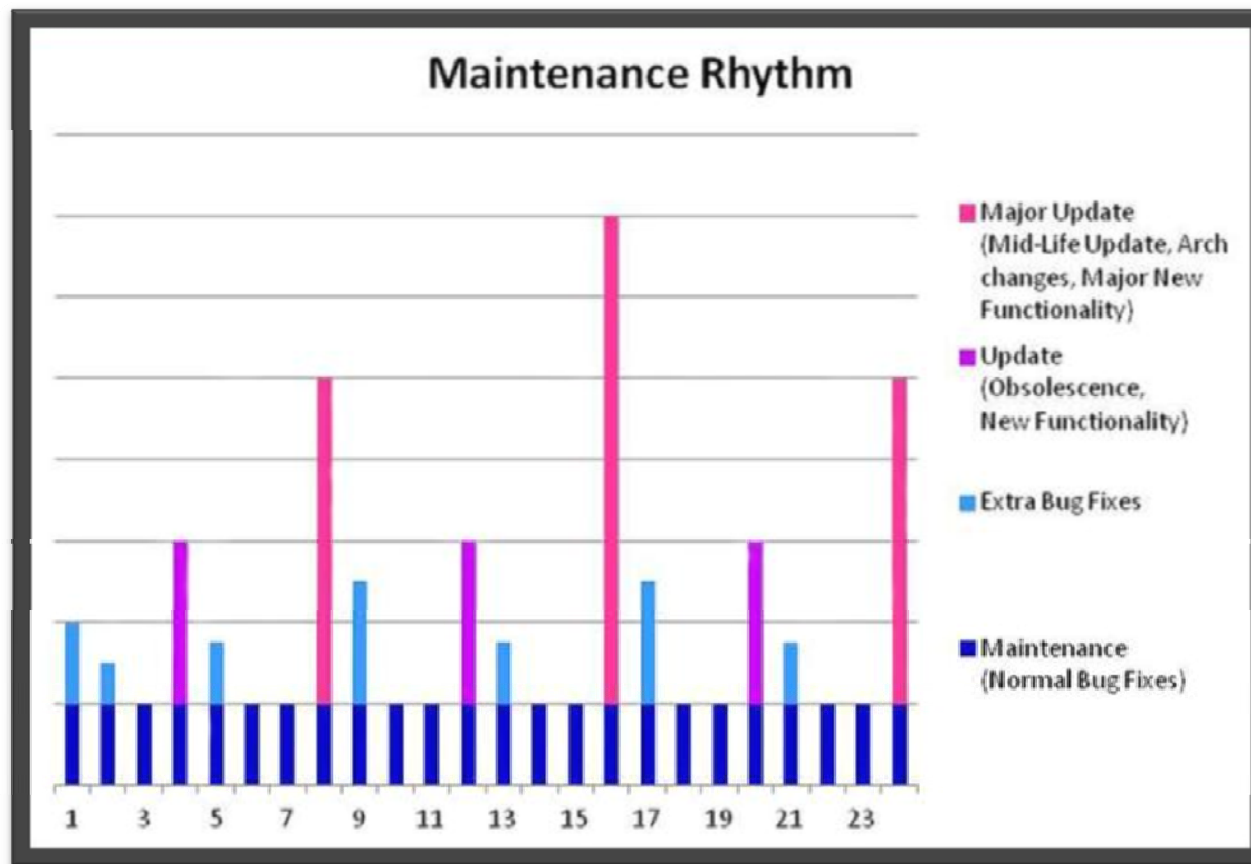
Integrated Cost Estimation Life Cycle Model

- ***Two key components***
- ***Software maintenance cost risk-uncertainty model***
 - ***Risk information model***
 - ***Risk interdependency analysis model***
 - ***Cost estimation application process model***
- ***Integrated life cycle cost allocation model***
 - ***Program software life cycle product-activity profile***
 - ***Post-development software releases***
 - ***Fielded software sustaining engineering support***
 - ***Software maintenance site infrastructure***
 - ***Maintenance resource-funding allocation model(s)***
 - ***Integrated cost estimation results***

Notional Software Maintenance Life-Cycle Cost Allocation Model



Program Software Maintenance Release Profile

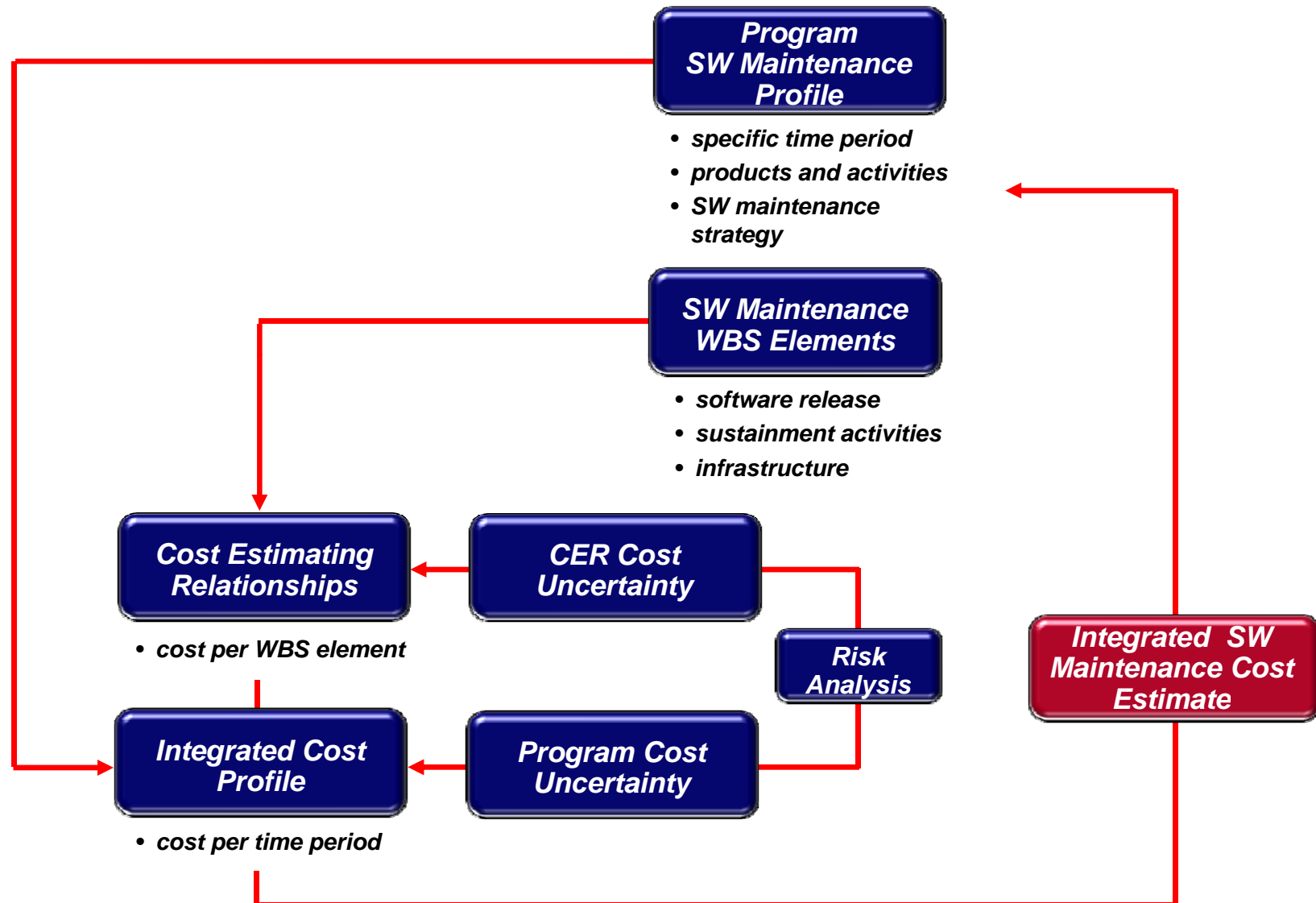


***Cycles are different - platform dependent
User needs drive release content***

Program Information Requirements

- ***System and software maintenance strategy***
- ***Included products and activities***
- ***Software release profile - PDSS/PPSS***
 - ***Release type and schedule (high level)***
 - ***Release profile stability***
 - ***User base - deployment strategy***
 - ***Configuration information***
 - ***Performing organizations***
 - ***Projected operational environment***
 - ***Program and software risk analysis***
 - ***Historical profile information (later)***

Practical Software and Systems Measurement



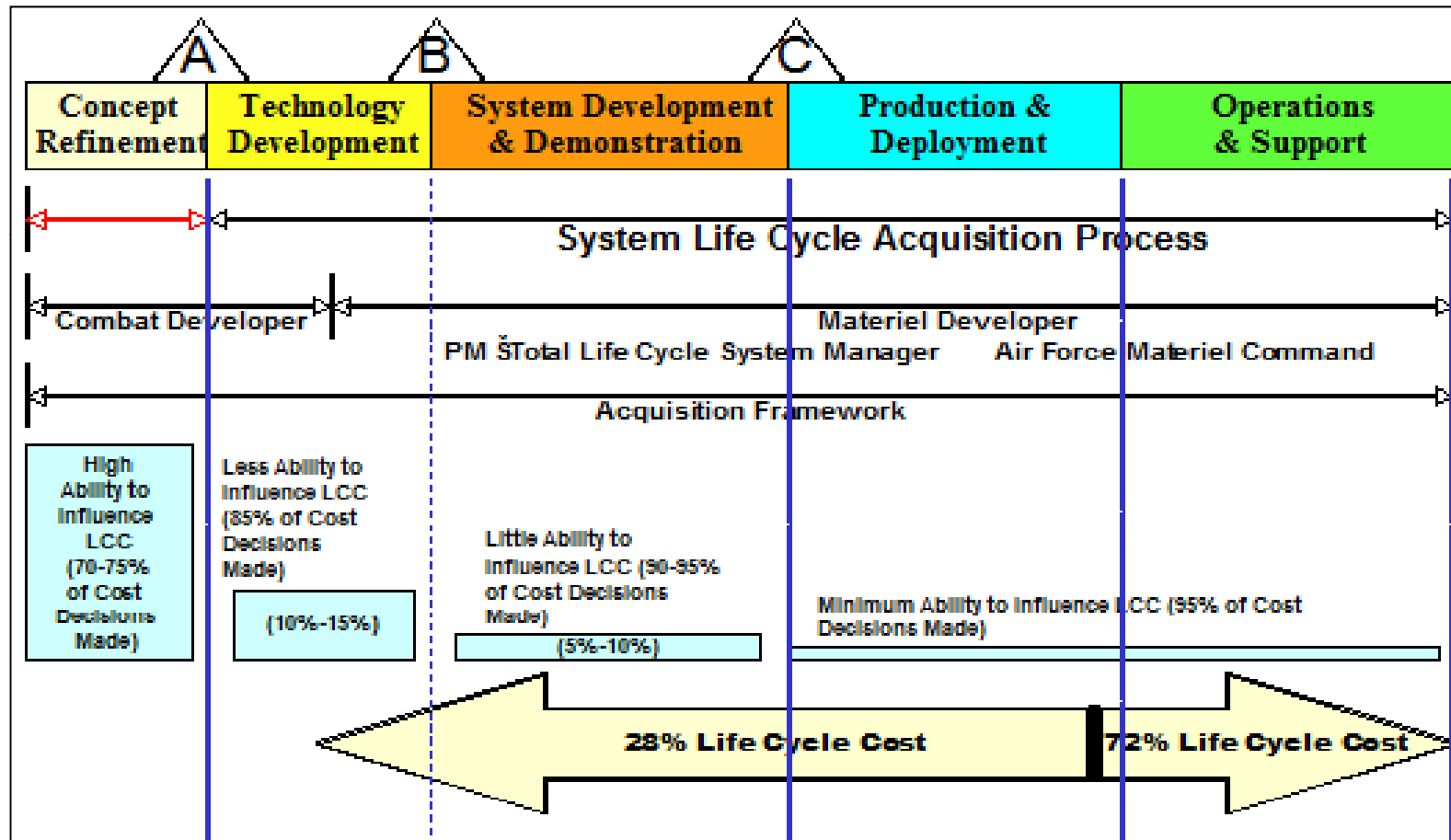
National Defense Authorization Act for Fiscal Year 2012 Implementation Guidance for Depot-Level Maintenance

- ***10 U.S.C. § 2460 - Definition of depot-level maintenance and repair has been changed (as of 05 April 2012) to state that software maintenance is defined as:***

"The repair, adaptive modifications or upgrades, change events made to operational software, integration and testing."

Interpretation/Guidance: The scope of this statute, as indicated by the title, is limited to depot-level maintenance and repair. Therefore, the language does not apply to hardware or software modifications that are not maintenance in nature. Specifically, modifications designed to enhance performance or add functional capability are not considered maintenance and may be excluded in the application of all depot maintenance statutory provisions, however, they may be included if the modifications are being executed consistent with past depot practices.

Acquisition Life-Cycle Model



10 USC 2366a states that:

“Milestone Decision Authority must certify that a determination of applicability of core depot-level maintenance and repair capabilities requirements has been made before a Major Defense Acquisition Program (MDAP) may receive Milestone A approval.

Milestone A

- **Interpretation/Guidance**: The scope of this analysis is limited to the determination whether the criteria detailed in 10 U.S.C. § 2464 apply to the weapon system or item of military equipment being procured. This analysis should be limited to determining whether the entire system, or particular subsets of the system, is considered to require core depot maintenance capability.
- The Cost Analysis Requirements Description (CARD) should have described or outlined a system sustainment concept, but details are not required for Milestone A.

10 USC 2366b states that:

“A major defense acquisition program may not receive Milestone B approval... until Milestone Decision Authority certifies: an estimate has been made of the requirements for core depot-level maintenance and repair capabilities, as well as the associated logistics capabilities and the associated sustaining workloads required to support such requirements .”

Requirements for LRIP:

The Secretary of Defense shall ensure that the detailed requirements for core depot-level maintenance and repair capabilities, as well as the associated logistics capabilities and the associated sustaining workloads required to support such requirements, have been defined.

Milestone B

- **The Cost Analysis Requirements Description (CARD) should now describe:**
 - **System sustainment concept**
 - **System logistics concept**
 - **Maintenance concept**
 - **Supply management concept**
 - **Transportation concept**
 - **Software maintenance concept**
 - **System training concept**
- **Develop Initial Capabilities Document (ICD) / Capability Development Document (CDD) / Capability Production Document (CPD)**
- **System Engineering Plan**
- **Life Cycle Sustainment Plan (DoD 5000.02) documenting the PM's plan for formulating, implementing and executing the sustainment strategy is required to achieving and maintaining the Sustainment KPP/KSAs needs to be developed as part of the official program/project Acquisition Strategy.**

Milestone B (Continued)

Initial Capabilities Document (ICD) / Capability Development Document (CDD) / Capability Production Document (CPD)

These documents are the sponsor's means to specify authoritative and testable, performance capabilities for the program. The ICD prefaces a system materiel decision and evolves into the CDD, which prioritizes KPP and subset KSA performance capability design and development parameters. The baseline CPD is finalized after the system level Critical Design Review and before Milestone C. In addition to supportability related KPP/KSAs, the ICD, CDD, and CPD narrative should also address the following:

- **System maintenance/support concepts and usage scenarios**
- **Operational and support environments. This should include the general support categories relative to the logistics support infrastructure (remote sites, organic depots, commercial facilities, air bases or ship yards, etc. without naming specific locations)**
- **Expected durations of support**
- **Support or maintenance effectiveness metrics and key enablers, such as diagnostics/prognostics**
- **Conditions conducive to joint sustainment and to performance-based support strategies.**

Milestone C



Source: U.S. Army

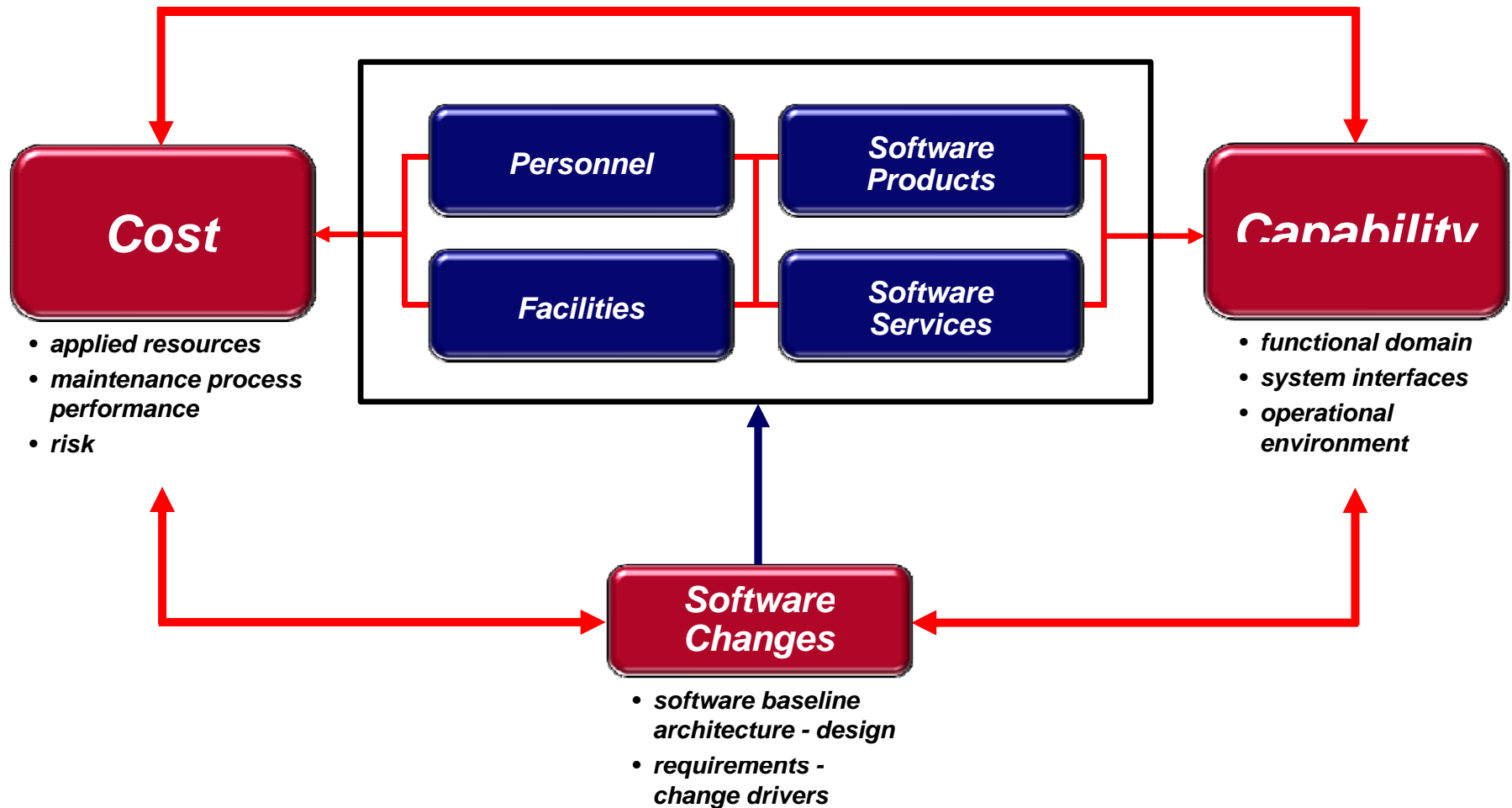
- **The Cost Analysis Requirements Description (CARD) is updated (and before Full Rate Production Decision Review)**
- **Initial Capabilities Document (ICD) / Capability Development Document (CDD) / Capability Production Document (CPD) are updated**
- **The Systems Engineering Plan is updated**
- **Life Cycle Sustainment Plan (DoD 5000.02) is updated**

Post Milestone C

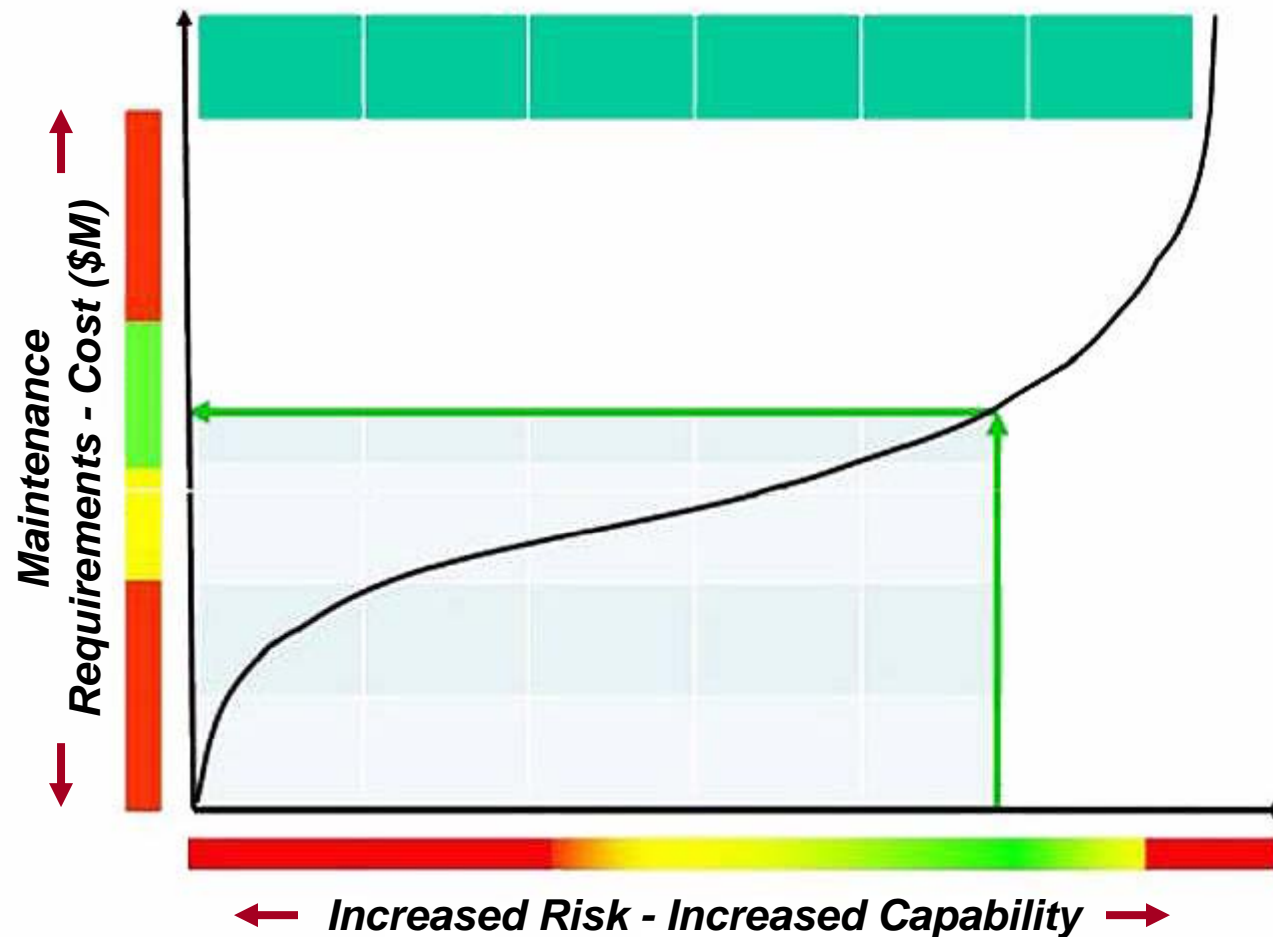


- **Life Cycle Sustainment Plan (DoD 5000.02) is updated at a minimum every 5 years, or when:**
 - **Subsequent Increments are approved and funded to reflect how the support strategy will evolve to support multiple configurations**
 - **Significant changes are required to the product support package to achieve the objective sustainment metrics including major support provider changes.**
- **If a decision has been made that a system will replace another, a Replaced System Sustainment Plan for the existing system is required that includes at least the budget estimates required to sustain the existing system until the new system assumes the majority of mission responsibility.**

Software Maintenance Cost-Capability Framework



How much should software maintenance cost?



"It's All About the Money", Dr. Chien Huo, CAPE, November 2011

Army Software Maintenance Study

- ***Sponsored by U.S. Army Office of the Deputy Assistant Secretary of the Army for Cost & Economics (ODASA-CE)***
- ***Collaborative environment - diverse perspectives***
- ***Army, Air Force, Navy, OSD, Industry participation***
- ***Initial focus on ACAT-1 weapons system software maintenance costs***
- ***Technical approach based on measurement and estimation best practices***

Contact Information

Jack McGarry

**U.S. Army RDECOM - ARDEC
Quality Engineering & System
Assurance
RDAR-QES-A
Building 62
Picatinny Arsenal, NJ 07806-
5000
(973) 724-7007
john.mcgarry@us.army.mil**

Dr. Robert Charette

**ITABHI Corporation
11609 Stonewall Jackson Drive
Spotsylvania, VA 22551-4668
(540) 972-8150
charette@itabhi.com**